

Atelier TEI-EPISTEMON
11/05/2015, CESR, Tours

Présents : Lou Burnard, Marie-Luce Demonet, Lauranne Bertrand, Mathieu Duboc, Sandrine Breuil
Ordre du jour : Audit corpus XML-TEI Epistemon, passage en revue des problèmes relevés dans le corpus, Mise à jour du manuel TEI-Renaissance et de l'ODD, génération du schema

Procédure Lou :

- feuille style 1 - copy > copier fichiers sans modification
 - feuille style 2 - cleanup > générer nouveau fichier
 - + comparaison fichiers > comptage des occurrences
- > modification ODD en parallèle > à joindre au Manuel

Toujours pointer vers URI

- feuille de style 3 > regarde toutes les occurrences et donne la liste des valeurs uniques

Langues

iso à 2 lettres et pas 3 > nous, **validé : mettre toujours 3 lettres > Lou l'intègre** à sa clean up

Idee glossaire MLD pour expliquer termes en langues régionales > encodage avec <distinct> pour substituer par un terme moderne > **à rediscuter plus tard avec Lou.**

ClassDecl

Taxinomie > ID et valeurs à valider / clarifier > voir liste valeurs envoyées par Lou > **liste à valider/"fermer" en interne BVH**

Corr / hand

Voir ex. Arrêts de Montaigne

Comment englober un bloc écrit par une main spécifique ?

Besoin d'un attribut @hand englobant > **cf. tei-L car discussion en cours**

Lou : @resp maintenant disponible, mais ptet pas suffisant car ambigu

div/ab

@corresp

Usage incorrect, faute de @hand. Lou nous indique qu'il est normalement possible d'utiliser @resp. **À confirmer**, pour l'instant tei_all ne le permet pas.

- Usage dans le Tory sur les <figure> doit être supprimé.

@xml:base = préfixe W3C pour se simplifier la vie (ne pas répéter une URL par ex.)

Qu'en fait-on ?

Si valeur ref1 = est supprimé

Si d'une autre forme = est transformé en sous éléments :

Si authority ou edition > transformé en elt fils ref > Lou fait pareil pour les bibl, authority, edition

Pour <graphic> : Pointe vers ressource propriétaire de l'image

URL vers la page de titre du fac-similé dans transcription

> **comportement à changer d'XTF** car URL déjà dans header > Lou le suppr dans page de titre. À voir de notre côté comment faire apparaître l'image via la transfo XSLT.

Pour foliotation restituée ? comment ?

<supplied> + @type ?

Pourrait n'être géré qu'à l'affichage ? non, on veut le garder dans le fichier !

- > utiliser @n du <pb> pas dans le <fw>
- > penser à mettre à jour script passage XML des Fs
- > LBe fait la mise à jour dans nos textes
- > Attention aux numérotations via @n déjà existantes

g @ref vides

- > à supprimer en attendant que l'on ajoute les liens systématiquement

@key

= normalement pointe vers ailleurs, peut donc être présents plusieurs fois dans le corpus ou dans le même texte. ex. des Arrêts de Montaigne

<name> et <persName>

= historique des BVH > évolution avec projet ReNom

- > à harmoniser partout <name> devient toujours <persName> > par Lou

+ fichier type à mettre à jour LBe

- > cas des groupes : sortir la liste et traiter au cas par cas...

- > cas de @key sur <author> = à supprimer car ça ne pointe vers rien (pers1, pers16 etc)

- > les @key de reNom seront précédés du préfixe URI bvh: = À répercuter dans le nettoyage de Marie dans la procédure ReNom.

<relatedItem> (header)

- À supprimer. @type = original est reporté sur la balise <bibl> dans les <surrogates>

Passage en revue du clean up de Lou

name, si @ID repasse sur @ref, si pas de # l'ajoute également sur @id des <div> et concatène pour créer id unique pareil pour <handnote>

pour **taxonomy**

@xml:id pour identifier le schéma > transformé en scheme

Pb avec **Bibl** quand figure plusieurs fois dans la liste des textes

- > soit on supprime les id des bibl, soit on les mets dans un fichier externe... à trancher

milestone (13 occurrences)

à supprimer ? LBe préfère les regarder pour vérifier si nécessaires.

- > Lou supprime le xml:id

zone

1 seul.. 1 test...

g

plusieurs ref ont valeur de target qu'il faut élargir = pointent vers des div > à changer par Lou

Idem pour le **CatRef**

Lou en identifie 3 : Corpus / version / unknown

Liste des @key utilisés mais qui ne sont pas utilisés

Attention répercussion sur chaîne de production ReNom avec ajout "bvh:" dans id
> [mettre à jour feuille de transformation](#) des textes en amont de l'import sous Renom

si resp n'est pas en pointeur = est transformé en pointeur
<resp> @ resp = redondance, suppr.
idem pour hand

Vérifier si "matière5" utilisé avant passage en catRef auto par Lou

@rendition > supprimés ! "car ils sont bêtes !"

@rend sur <ab> à conserver !

Bibl dans header > @surrogates seulement pour autres numérisations...

Mais où mettre l'url vers notre fac-similé de référence ?

> [@ à ajouter sur bibl par ex. type="original"](#) [moi je préférerais "exemplaire_reference" ou terme plus proche de ça...]
>

identifiant sur identifiant ? ...

recordHist mal utilisé !

listBibl dans surrogates

> important : en gros on déplace les références bibl de l'original et des exemplaires dans surrogates, qu'il y ait une <ref> vers version num ou pas d'ailleurs. Chaque bibl a un @xml:id unique donc à voir si on externalisera pas cette bibliographie dans un fichier à part. Ce qui fera que dans le surrogates ne demeureront que des pointeurs vers les <bibl> de cette liste.
> par contre, @xml:id sur msDesc pas sur msIdentifier pour l'original.

restore

ex. dans Montaigne

S'il est damage, logique qu'on le restaure, donc pas besoin de la préciser

Lou suggère de ne pas mettre le damage, juste le restaure car on veut mettre en évidence l'intervention de Montaigne

Cas des

- **resp** = toujours pour une personne
- et **sources** = édition de référence, un livre

> [à vérifier dans textes + mettre à jour les XSLT](#)

Changement de la TEI pour **@unit**

> [Lou le change partout](#)

@version sur tei n'existe plus / n'est pas remplacé

> à supprimer dans les premiers textes publiés... [Lou](#).

Typologie / Taxinomie

Idée groupe de travail CAHIER sur typologie commune générique/espèce

Mélange des types / genres / sous-types / sous-genres

> différencier ce qui est du genre dans <header> et se qui caractérise des éléments (div, etc.)

@keyword = manière de décrire les catégories avec typologie fermée = décrit le texte entier

1. keyword : peuvent être définies dans le header et pointer dans le texte sur des parties/divisions > fichier externe pour maintenance
2. ou liste fermée dans le schema pour attribut type sur <div> > avantage : validation avec schéma plus facile Solution la plus viable.

Tous **@type** et **@subtype** à harmoniser par nous > Lou nous sort la liste > une fois validé, seront intégrés au schema

corr @resp

> à contrôler et à passer en **@source**, s'il y a une personne effectivement responsable alors **@resp**. Distinguer **@source** de **@resp** dans l'ensemble des textes d'ailleurs.

> Lou traite l'ensemble > à déclarer dans le header

Cas de pelletier traducteur d'une <div> >> à mettre en <note> ? > sera supprimé de **@corresp** par Lou

Traduction = 1 genre à ajouter dans typologie

gap (24 occurrences) = un trou sans rien d'autre

Dans EB (folio 7) > partie endommagée par tâches d'encre > pourquoi un pointeur ? A supprimer.

On ajoutera un supplied avec **@source** au besoin

Regarder l'ensemble des **bibl** et voir si fichier externe des sources >> id uniques :)

> expérience à faire par Lou

listPers = personnes mentionnées

ex. Lettre Eli Vinet > M. Loisel > xml:id déclaré dans <header>

Resp1... resp6... à supprimer

>> **NETTOYER le FICHIER-TYPE de toute URGENCE !! o_0'**

> identification des liens s'est bien passé (sauf qq fautes gérables)

@who plus complexes et problématiques

- **Pour théâtre** :

Pb avec **castList** = phénomène régulier de liste de personnages donnée > à corriger dans tous les textes (comme cela a été fait dans le support de formation) > **castList** = seulement vraie liste complète au début.

Ajoute-t-on une **castList** artificielle si absente ? oui, dans le header avec **listPerson @ who**.

Si déjà existante, la duplique-t-on dans le header ? oui, car permet enrichissement et modernisation des formes.

said @who : Pb d'espaces à corriger + pb de # ajoutés

said @resp / supplied @resp : mêmes problèmes

Répertorier les personnages dans header > dans **listPerson**

lb @rend=hyphen

Usage curieux dans notre corpus car toujours des blancs avant (blancs d'indentation) > scripts les nettoient

> voir si à transformer en <lb break="no"/> ou pas.

Cas du **editor role=traducteur** : à déclarer dans <respStmt> ?

Planning

- Lou enverra un second rapport de validation du corpus
- Génération du Manuel à partir de l'ODD > parties manquantes à rédiger + intégrer ODD à la fin + mise en page html à faire ! = prends du temps > [Lou propose d'aider Lauranne](#) sur Manuel

(transformation en 2 étapes !)

> idéal serait pour Lauranne de n'avoir plus que la CSS à modifier

> puis PDF à faire ensuite

- Validation du nouveau corpus ? on doit encore les contrôler en interne...

Lou dispo deux premières semaines de juin

Envoi par Lou du Manuel + textes à valider par nous vers 15 juillet au plus tard

Lou envoi facture 1800€ pour mise en paiement

Rencontre prépa formation TEI avec Lou et Alexei : semaine du 15 septembre ? (pas dispo avant ni après) > 2 jours ? **A fixer début juillet** ? Atelier Manuel + validation corpus + formation TEI + TXM ?

Messages>

<ERROR>target points to non-existent ??

(teiCorpus[1]/TEI["B330636101_S1238_encodage_EG_Livrel_tei"]/teiHeader[3]/fileDesc[4]/sourceDesc[5]/msDesc[6]/additional[7]/surrogates[8]/bibl[9]/relatedItem[10]/ref[11]/) </ERROR>

<ERROR>new points to non-existent MDG

(teiCorpus[1]/TEI["B330636101_S1238_encodage_EG_Livrel_tei"]/text[3]/body[4]/div["B330636101_S1238_encodage_EG_Livrel_n5"]/div["B330636101_S1238_encodage_EG_Livrel_n5.24"]/p[7]/add[8]/handShift[9]/) </ERROR>

<ERROR>target points to non-existent add73r_3

(teiCorpus[1]/TEI["B330636101_S1238_encodage_EG_Livrel_tei"]/text[3]/body[4]/div["B330636101_S1238_encodage_EG_Livrel_n5"]/div["B330636101_S1238_encodage_EG_Livrel_n5.28"]/p[7]/add[8]/add[9]/metamark[10]/) </ERROR>

<ERROR>ref points to non-existent it

(teiCorpus[1]/TEI["B360446201_B343_1_tei"]/teiHeader[3]/fileDesc[4]/sourceDesc[5]/msDesc[6]/additional[7]/adminInfo[8]/recordHist[9]/source[10]/listBibl[11]/bibl[12]/title[13]/) </ERROR>

<ERROR>who points to non-existent Rissan

(teiCorpus[1]/TEI["B360446201_THIA63_1_tei"]/text[3]/body[4]/div["B360446201_THIA63_1_n4"]/p[6]/said[7]/) </ERROR>

<ERROR>who points to non-existent Regnard

(teiCorpus[1]/TEI["B360446201_THIA63_1_tei"]/text[3]/body[4]/div["B360446201_THIA63_1_n4"]/p[6]/said[7]/) </ERROR>

<ERROR>who points to non-existent Rissan

(teiCorpus[1]/TEI["B360446201_THIA63_1_tei"]/text[3]/body[4]/div["B360446201_THIA63_1_n4"]/p[6]/said[7]/) </ERROR>

<ERROR>who points to non-existent Rissan

(teiCorpus[1]/TEI["B360446201_THIA63_1_tei"]/text[3]/body[4]/div["B360446201_THIA63_1_n4"]/p[6]/said[7]/) </ERROR>

</Messages>

aute de validite:

B330636101_D55869_tei.xml:146:94:

l'attribut @type a ete remplace sur biblScope par @unit

Il n y a que 3 occurrences de <handShift> dans tous ces fichiers, tous dans Montaigne.

2/3 sont (correctement) pointeurs #MDG et #MDM, mais il n y a aucun <handDesc xml:id="MDG"> correspondant.

Le 3eme manque de # et donc ne fonctionne pas

Tu peux m'expliquer l'intention derriere l'usage de <metamark> dans ces textes?

Par exemple:

```
<supplied resp="#MD">
<add>
<metamark target="#add6r_4"></metamark>
</add>
</supplied>
  <add type="marginale" place="margin-left">
    <lb/>
    <metamark></metamark> La <subst>
    <del>peine <choice>
      <orig>suiuit</orig>
      <reg>suivit</reg>
    </choice>
    </del>
    <add place="top">fortune</add>
  </subst> quelques annees apres
```

a) L'attribut @target devrait cibler qqchose, mais je ne trouve aucun element avec un identifiant "add6r_4"

b) le caractere utilise pour représenter la fleche (?) s'affiche comme un blanc dans oXygen : c'est quoi exactement?

c) <supplied resp="#MD"> m'indique que ou bien c'est MD qui a decide qu'il s'agit ici d'un "supplied" , ou que c'est MD qui a fourni ce contenu. Mais pourquoi le <add> en plus?

d) et puisque nous y sommes, pour @type="marginale" ET @place="margin-left"? Un ajout place dans les marges est forcément marginale n'est ce pas?

<metamark> : on marque ici le signe d'insertion utilisé par Montaigne, dont la forme peut varier mais qu'on a remplacé par un signe unique pour la lisibilité et l'affichage.

a) C'est qu'il y a un problème dans l'encodage, tout target doit cibler une add précise.

b) La flèche ? Le signe< ? C'est pour "insertion", mais il s'affiche dans oxygen de mon côté vu qu'on a mis le caractère hexadecimal

c) C'est qu'il a fourni le contenu du supplied, il est donc responsable de la proposition. Un <add> en plus parce que le signe est une addition de Montaigne, qu'on restitue.

d) Alors @type pour pouvoir faire un corpus dans TXM de ces additions marginales, pour pouvoir cibler toutes les additions marginales qu'elles soient à droite ou à gauche. Même si c'est un peu redondant.

Tu me diras pour github si tu récupères mes mises à jour ou pas hein ;)

b) OK, pour l'affichage du signe \wedge je ne m'inquiète pas, c'est une question de police d'affichage, clairement.

c) Je n'arrive toujours pas à comprendre l'usage `<supplied> <add> ... </add> </supplied>` :
`<supplied>` veut dire qu'il manque qqch dans la version qu'on transcrit, qu'on a ajoutée. Mais ici, ce qu'on ajoute est un ajout ! Si c'est un ajout, forcément il ne manque pas dans l'original... ça me fait mal à la tête

a) Si tu dis que le `@target` ici est fautif, on est d'accord. Comment corriger, dans ce cas ?

d) Tu veux dire que TXM est tellement bête qu'il n'arrive pas à classer ensemble plusieurs valeurs avec des codes différents ? Tsk tsk. Pour moi, la redondance est un péché mortel ... ainsi que la variation inutile (tous les `<add>` ne sont pas types !)

Pour github ... c'est vraiment un mystère. J'ai récupéré les versions que tu avais envoyées sur l'ancien site, et je les ai copiées dans mon dépôt actuel. Mais si tu as fait des modifications ensuite je ne les vois pas. Tu peux me fournir avec un exemple ?

c) Eh bien on a quand même à la base une addition de Montaigne à l'imprimé. Addition rognée qu'on restitue. Donc c'est quand même une addition restituée non ? Moi aussi ça me file mal à la tête ! Ça fait bizarre tu as raison.

a) Hmm... Eh bien en effet, ça va être compliqué de tous les vérifier à la main.

d) Actuellement ça me simplifie la vie. Mais peut-être que c'est moi qui ne l'utilise pas correctement.

Github : Je ne sais pas où est-ce que je livre. Regarde par ici si tu peux :

<https://github.com/laurannebp/tei-fr>

Je pense que je fais des livraisons avec moi-même. J'ai fait "fork" dans github, car si je prenais l'url de ton dépôt à toi, ça me renvoyait une erreur.

Je te mets en copie mes dernières livraisons.

Bonne soirée Lou !

(c) Je comprends toujours pas ! Le contenu de `<supplied>` devrait être une partie fournie par l'éditeur. Dans ce cas, il contient que le metamark. Ce qui implique l'absence de ce metamark dans l'original. Il est suivi d'un deuxième metamark. S'agit-il donc de 2 marks, dont le premier est fourni par `#MD` ? Et pourquoi ? Le premier des marks cible vers quelque chose, mais pas le dernier. Vers quoi ? Et pourquoi pas le 2ème ?

(a) on peut les vérifier s'ils sont présents !

Merci de tes versions. Je regarde ça après dîner... malheureusement je ne comprends pas Git non plus : je viens de te faire "collaborator" et il me dit que tu as déjà le droit de faire des "push" sur le repo. Mais puisque tu l'as "forké" je suppose que moi je dois accepter tes modifications.

Si tu fais un update tu devrais voir maintenant les versions que tu viens de m'envoyer, en tout cas.

c) Le premier metamark restitué cible le second metamark. En fait Montaigne insère un signe pour pointer vers une autre addition qu'il continue quelque part dans les marges de son livre.

a) Oui, c'est vrai. Et les lister aiderait à corriger ceux qui manquent.

Alors pour Github, je ne comprends pas trop mais j'ai réussi à extraire ton repository et à livrer dessus. Mon fork ne servait à rien et a peut-être fichu le bazar pour le coup. Toujours est-il que je livrerai dessus à présent. Tu devrais voir mon test du matin.

B751131011_RES_P_Y2_1349_tei

```
<pb n="34"/>
<fw type="pageNum" place="top-left">[34]</fw>
<lb/>quadernes, ternes, double deux. Les autres
<lb/>
<app>
  <lem>escartelets</lem>
  <rdg wit="source= #FR2156">escoullettez</rdg>
</app>
```

=> CORRIGÉ

B751131011_Y22789_tei

<sp who="Mar. Ficinus" rend="runon">

L'attribut @who est 1:n IDREF, donc ca veut dire qu'il y a 2 personnes qui parlent ("Mar.", et "Ficinus")

Histoire de blancs encore: dans

<said who="#Regnard Rissan">Par

<said who="#l_Ardelier Regnard Rissan">

(B360446201_THIA63_1_tei)

il faut ajouter un # pour chacun des liens, s'il s'agit de plusieurs personnes parlant en même temps.
Ou supprimer le blanc s'il n'y a qu'un locuteur.

Oups.... B360446201_B343_1_tei

<title ref="it">Oeuvres romanesques (les cinq livres de Pantagruel)</title>,

devrait etre rend="it", je suppose

Qqchose un peu plus interessante que ces histoires de pointeur ... (UN PEU)

J'attache un template (autogenere par la fds divTypes.xsl) qui permet de controller/corriger tous les valeurs div/@type attestés dans le corpus

Je propose que tu regarde bien les valeurs utilisées et

(a) si c'est bon, enleve entierement du template le "xsl:when"

(b) sinon, modifier la valeur propose

par ex

tu vois

```
<xsl:when test="bidule">bidule</xsl:when> <!-- 3 -->
```

si c'est bien, tu enleves cette ligne

si tu veux que cette valeur soit remplace par "machin" tu modifie la ligne en

```
<xsl:when test="bidule">machin</xsl:when> <!-- 3 -->
```

OK?

tu me renvoie le resultat et je l'integre dans mon procedure de cleanup

Si ca te va, je ferai ensuite pareil pour d'autres attributs qu'on souhaite normaliser...

Quelques commentaires :

```
<xsl:template match="tei:add/@xml:id"/>
```

Pourquoi cette suppression ? Chez Montaigne, les @xml:id sur les <add> sont essentiels.

D'abord, Montaigne ne figurait pas dans le corpus que depuis hier. Dans la reste du corpus il n'y a que deux <add>s qui portent un @xml:id (avec les valeurs "add1" et "add2" respectivement), et aucun élément ciblant la-dessus. Donc j'ai enlevé.

Ah oui. Pour le coup, il faut le laisser et plutôt enlever à la main ceux qui ne pointent vers rien ?

```
<xsl:template match="tei:ab/@rend"/>
```

Pourquoi ? Les <ab> ne peuvent pas avoir de stylage ?

Si, bien sûr. Mais dans le corpus je vois 1022 <ab>

8 n'ont pas de @rend, 3 ont @rend="center" et tout le reste @rend="runon".

Alors il me paraît que la rendition par défaut de <ab> est "runon", et qu'il serait plus simple de signaler les exceptions.

Mais tu as raison de me poser la question!

Hmm. Si 8 n'ont pas de @rend, ce sont eux qui malgré tout qui sont les <ab> par défaut.

le @runon c'est pour le théâtre je pense.

Je préfère qu'on le laisse ce @rend, par exemple, dans les 47 arrêts que j'ai rajouté, on a des <ab> non stylés, majoritairement.

Et qui ont bien des retours à la ligne, pas du runon (qui est en fait là pour annuler un retour à la ligne par défaut).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:template xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:tei="http://www.tei-c.org/ns/1.0"
  match="tei:*/@type">
  <xsl:attribute name="rend">
    <xsl:choose>
      <xsl:when test=".='center'">center<!--1--></xsl:when>
      <xsl:when test=".='hidden'">hidden<!--1--></xsl:when>
      <xsl:when test=".='hyphen'">hyphen<!--9--></xsl:when>
      <xsl:when test=".='it'">it<!--2--></xsl:when>
      <xsl:when test=".='medium'">medium<!--1--></xsl:when>
      <xsl:when test=".='runon'">runon<!--2--></xsl:when>
      <xsl:otherwise>
        <xsl:value-of select="."/>
      </xsl:otherwise>
    </xsl:choose>
  </xsl:attribute>
</xsl:template>
```

Oui, et normalement c'est "it" (donc il faudrait remplacer italique par it)
center (donc idem pour centre).

Pour info :

```
<xsl:template name="rend">
  <xsl:choose>
    <xsl:when test="@rend='center' or @rend='head_columns_center'">
      <xsl:attribute name="style">text-align:center;display:block;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='it' or @rend='italic' or @rend='ital'">
      <xsl:attribute name="style">font-style:italic;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='center_it'">
      <xsl:attribute name="style">font-style:italic;text-align:center;display:block;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='small_it'">
      <xsl:attribute name="style">font-style:italic;font-size:75%;display:block;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='bottom'">
      <xsl:attribute name="style">font-size:12px;vertical-align:sub;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='top' or place='top'">
      <xsl:attribute name="style">font-size:12px;vertical-align:super;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='sup'">
      <xsl:attribute name="style">font-size:12px;vertical-align:super;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='columns_center_it'">
      <xsl:attribute
name="style">width:auto;float:left;margin-left:100px;font-style:italic;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='right'">
      <xsl:attribute name="style">float:right;display:block;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='alinea'">
      <xsl:attribute name="style">margin-left:10mm;display:block;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='bold'">
      <xsl:attribute name="style">font-weight:bold;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='justify'">
      <xsl:attribute name="style">text-align:justify;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='left'">
      <xsl:attribute name="style">text-align:left;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='novspace'">
      <xsl:attribute name="style">margin-top:0px;</xsl:attribute>
    </xsl:when>
    <xsl:when test="@rend='code'">
```

```

    <xsl:attribute name="style">font-weight:bold;font-family:monospace;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='sc'">
    <xsl:attribute name="style">font-variant:small-caps;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='style'">
    <xsl:attribute name="class">font-style:normal;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='runon'">
    <xsl:attribute name="style">display:inline;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='ul' or @rend='underline'">
    <xsl:attribute name="style">text-decoration:underline;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='overbar' or @rend='overline'">
    <xsl:attribute name="style">text-decoration:overline;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='strike' or @rend='line-through'">
    <xsl:attribute name="style">text-decoration:line-through</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='expanded'">
    <xsl:attribute name="style">letter-spacing:0.15em;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='small'">
    <xsl:attribute name="style">font-size:75%;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='large'">
    <xsl:attribute name="style">font-size:150%;line-height:120%;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='larger2'">
    <xsl:attribute name="style">font-size:400%</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='smaller'">
    <xsl:attribute name="style">font-size:85%</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='indent'">
    <xsl:attribute name="style">text-indent:5%;margin-top:-2.5%;margin-bottom:0;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='larger'">
    <xsl:attribute name="style">font-size:200%;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='calligraphic'">
    <xsl:attribute name="style">font-family:cursive;</xsl:attribute>
</xsl:when>
<xsl:when test="@rend='gothic'">
    <xsl:attribute name="style">font-family:fantasy;</xsl:attribute>
</xsl:when>
</xsl:choose>
</xsl:template>

```

Interessant!

Mais pourquoi donc continuer d'utiliser @rend?

Ne serait-il pas plus simple d'utiliser directement @style avec les valeurs indiquées?

Ou (encore mieux) utiliser @rendition ?

par ex

```
<foo rendition="#gothic">  
qui pointe sur  
<rendition xml:id="gothic">font-family:fantasy</rendition>
```

Oui, @style est globalement disponible depuis longtemps (voir <http://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ref-att.global.rendition.html> et <http://www.tei-c.org/release/doc/tei-p5-doc/fr/html/ST.html#STGAre>)

Le gros avantage de @rend est qu'il vous permet d'utiliser des codes arbitraires pour la mise en forme -- et c'est aussi son gros désavantage, parce que ces codes ne sont ni validables ni traitables par les outils déjà existants.

Le gros avantage de @style (avec CSS) est qu'il vous permet de définir exactement les aspects visuels en ayant la possibilité de voir cela réalisé sans travail par un navigateur; le gros désavantage c'est que la langue CSS est un peu difficile à gérer (et très verbeuse) au niveau des éléments individuels; vous perdez d'ailleurs la possibilité de faire des recherches ou analyses simples, parce qu'il faut d'abord traiter toute l'hierarchie CSS en vigueur pour savoir par exemple si tel ou tel élément soit en italique.

Le gros avantage de @rendition est qu'il combine les gros avantages des deux autres, et surajoute un niveau de validation et contrôle. Le désavantage est qu'il faut pré-définir tous les renditions qu'on prévoit dans un document et maintenir cet ensemble de déclarations, mais c'est pareil pour plusieurs autres aspects (catRef, div@type, @resp etc)

TEI Boilerplate utilise @rendition directement

N'oubliez pas aussi que l'objectif de tous ces attributs serait de décrire l'affichage original de la source, et non pas un affichage souhaité. Donc ils servent à documenter autant qu'à traiter...

(J'aime bien ton usage de "mitiger" qui m'a coûté quelques minutes de recherche internet... en anglais on dirait "I have mixed feelings" ?)

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:template xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:tei="http://www.tei-c.org/ns/1.0"
  match="tei:div/@type">
  <xsl:attribute name="type">
    <xsl:choose>
      <xsl:when test=".='Lettre'">Lettre<!--1--></xsl:when>
      <xsl:when test=".='acte'">acte<!--13--></xsl:when>
      <xsl:when test=".='ajout_ms'">ajout_ms<!--1--></xsl:when>
      <xsl:when
test=".='alphabet'">alphabet<!--16--></xsl:when>
      <xsl:when
test=".='anterotique'">anterotique<!--1--></xsl:when>
      <xsl:when test=".='argument'">argument<!--2--></xsl:when>
      <xsl:when
test=".='au_lecteur'">au_lecteur<!--9--></xsl:when>
      <xsl:when
test=".='avertissement'">avertissement<!--1--></xsl:when>
      <xsl:when test=".='ballade'">ballade<!--16--></xsl:when>
      <xsl:when test=".='blason'">blason<!--40--></xsl:when>
      <xsl:when test=".='carte'">carte<!--1--></xsl:when>
      <xsl:when test=".='chanson'">chanson<!--44--></xsl:when>
      <xsl:when test=".='chant'">chant<!--3--></xsl:when>
      <xsl:when
test=".='chapitre'">chapitre<!--938--></xsl:when>
      <xsl:when
test=".='colophon'">colophon<!--13--></xsl:when>
      <xsl:when
test=".='complainte'">complainte<!--4--></xsl:when>
      <xsl:when
test=".='conclusion'">conclusion<!--1--></xsl:when>
      <xsl:when
test=".='contemplation'">contemplation<!--1--></xsl:when>
      <xsl:when
test=".='contreblason'">contreblason<!--23--></xsl:when>
      <xsl:when test=".='couplet'">couplet<!--2--></xsl:when>
      <xsl:when test=".='debat'">debat<!--1--></xsl:when>
      <xsl:when
test=".='declaration'">declaration<!--5--></xsl:when>
      <xsl:when
test=".='dedicace'">dedicace<!--14--></xsl:when>
      <xsl:when test=".='dessein'">dessein<!--69--></xsl:when>
      <xsl:when
test=".='discours'">discours<!--71--></xsl:when>
      <xsl:when test=".='distique'">distique<!--1--></xsl:when>
      <xsl:when test=".='dizain'">dizain<!--6--></xsl:when>
    </xsl:choose>
  </xsl:attribute>

```

```

        <xsl:when test=".='douzain'">douzain<!--1--></xsl:when>
        <xsl:when test=".='eglogue'">eglogue<!--3--></xsl:when>
        <xsl:when test=".='elegie'">elegie<!--6--></xsl:when>
        <xsl:when test=".='eloge'">eloge<!--1--></xsl:when>
        <xsl:when test=".='embleme'">embleme<!--46--></xsl:when>
        <xsl:when
test=".='entreprise'">entreprise<!--4--></xsl:when>
        <xsl:when
test=".='epigramme'">epigramme<!--5--></xsl:when>
        <xsl:when
test=".='epitaphe'">epitaphe<!--16--></xsl:when>
        <xsl:when test=".='epitre'">epitre<!--34--></xsl:when>
        <xsl:when test=".='errata'">errata<!--8--></xsl:when>
        <xsl:when test=".='excipit'">excipit<!--1--></xsl:when>
        <xsl:when test=".='exemple'">exemple<!--8--></xsl:when>
        <xsl:when
test=".='glossaire'">glossaire<!--1--></xsl:when>
        <xsl:when test=".='histoire'">histoire<!--1--></xsl:when>
        <xsl:when test=".='huitain'">huitain<!--2--></xsl:when>
        <xsl:when
test=".='institution'">institution<!--1--></xsl:when>
        <xsl:when
test=".='introduction'">introduction<!--1--></xsl:when>
        <xsl:when
test=".='lamentation'">lamentation<!--1--></xsl:when>
        <xsl:when test=".='livre'">livre<!--30--></xsl:when>
        <xsl:when test=".='madrigal'">madrigal<!--1--></xsl:when>
        <xsl:when
test=".='notes_manuscrites'">notes_manuscrites<!--1--></xsl:when>
        <xsl:when
test=".='nouvelle'">nouvelle<!--90--></xsl:when>
        <xsl:when test=".='ode'">ode<!--116--></xsl:when>
        <xsl:when test=".='odes'">odes<!--1--></xsl:when>
        <xsl:when test=".='oeuvres'">oeuvres<!--1--></xsl:when>
        <xsl:when
test=".='oeuvres_de_jeunesse'">oeuvres_de_jeunesse<!--1--></xsl:wh
en>
        <xsl:when test=".='oraison'">oraison<!--1--></xsl:when>
        <xsl:when
test=".='page_de_titre'">page_de_titre<!--3--></xsl:when>
        <xsl:when test=".='plan'">plan<!--1--></xsl:when>
        <xsl:when test=".='poeme'">poeme<!--17--></xsl:when>
        <xsl:when test=".='point'">point<!--1--></xsl:when>
        <xsl:when test=".='postface'">postface<!--2--></xsl:when>
        <xsl:when test=".='preface'">preface<!--15--></xsl:when>
        <xsl:when

```

```

test=".='privilege'">privilege<!--23--></xsl:when>
    <xsl:when
test=".='prologue'">prologue<!--13--></xsl:when>
    <xsl:when
test=".='pronostication'">pronostication<!--2--></xsl:when>
    <xsl:when test=".='quatrain'">quatrain<!--3--></xsl:when>
    <xsl:when test=".='quintil'">quintil<!--1--></xsl:when>
    <xsl:when test=".='recette'">recette<!--1--></xsl:when>
    <xsl:when test=".='registre'">registre<!--2--></xsl:when>
    <xsl:when
test=".='remontrance'">remontrance<!--1--></xsl:when>
    <xsl:when test=".='reponse'">reponse<!--4--></xsl:when>
    <xsl:when test=".='resume'">resume<!--1--></xsl:when>
    <xsl:when test=".='rondeau'">rondeau<!--70--></xsl:when>
    <xsl:when test=".='septain'">septain<!--1--></xsl:when>
    <xsl:when
test=".='sommaire'">sommaire<!--26--></xsl:when>
    <xsl:when test=".='sonnet'">sonnet<!--220--></xsl:when>
    <xsl:when test=".='sonnets'">sonnets<!--2--></xsl:when>
    <xsl:when test=".='stance'">stance<!--1--></xsl:when>
    <xsl:when test=".='table'">table<!--6--></xsl:when>
    <xsl:when
test=".='table_des_auteurs'">table_des_auteurs<!--2--></xsl:when>
    <xsl:when test=".='tableau'">tableau<!--12--></xsl:when>
    <xsl:when test=".='traite'">traite<!--20--></xsl:when>
    <xsl:when test=".='vers'">vers<!--6--></xsl:when>
    <xsl:when
test=".='vers_lyriques'">vers_lyriques<!--1--></xsl:when>
    <xsl:otherwise>
        <xsl:value-of select="."/>
    </xsl:otherwise>
</xsl:choose>
</xsl:attribute>
</xsl:template>

```


a ligne 8531 je lis:

elle va toute seule, elle ne fait que traîne<damage>r</damage>
<restore>r</restore> & languir:

Ce qui implique qu'il y a d'abord un "r" endommagé, et puis un deuxième "r" restauré par l'éditeur.
C'est bien ça? Sinon, on peut supprimer le deuxième: évidemment, s'il est endommagé, sa présence dans la transcription implique une restauration.

ligne 8310 <fw type="catch" place="bot_right">demeu-</fw>

partout ailleurs c'est "bot-right". Ca devrait etre modifie, je crois...

ligne 75076 l'element <w> est utilise (uniquement ici) pourquoi?

Tous les metatag/@target dans ce texte ont une valeur de forme #addxxx

MAIS il n'existe nulle part aucun element portant le xml:id correspondant. A quoi ca peut servir donc?

Le xxx semble etre compose d'une indication de page suivi d'un numero de sequence. Si c'est bien ca, on pourrait utiliser une code magique "bvh:addxxx" je suppose. Mais la diese impose des obligations!

Dans la <msDesc> du texte Ms629, je trouve

```
<surrogates>
  <bibl>
    <relatedItem type="original">
      <ref target="Ms629">Bibliothèques Virtuelles Humanistes</ref>
    </relatedItem>
  </bibl>
</surrogates>
```

Qu'est-ce que ça veut dire?

Il manque le # devant ms629.

Ça doit pointer sur la description de l'exemplaire dans msdesc

En ce qui concerne ...

@xml:base sur les <bibl> et <authority> est transformé dans un élément enfant <ref>

@xml:base sur les <graphic> je ne sais pas quoi faire: par ex

```
<figure>
```

```
  <graphic url="XUVA_Gordon1578_L47.jpg"
```

```
xml:base="http://iris.lib.virginia.edu/speccol/gordon/gordonimages/Gordon1578\_L47/">
```

```
  <figDesc>Courtesy of Gordon Collection</figDesc>
```

```
</figure>
```

ce qui implique que je trouverai le graphique souhaité à l'adresse

http://iris.lib.virginia.edu/speccol/gordon/gordonimages/Gordon1578_L47/XUVA_Gordon1578_L47.jpg

mais il n'est pas là.

Egalement, dans cet exemple

```
<figure>
```

```
  <graphic url="B452346101_H6513.jpg"
```

```
xml:base="http://www.bvh.univ-tours.fr/Consult/index.asp?numfiche=851">
```

```
  <figDesc>Bibliothèque Virtuelles Humanistes</figDesc>
```

```
</figure>
```

je suppose que le @xml:base pourrait être un ref complémentaire, et que le jpg soit qqpart disponible?

B693836101_346632_tei.xml contient une passage curieuse

```
<foreign xml:lang="ell">
  <choice>
    <orig>
      <lb/>εις το αυτο.
      <lb/>Ζησεται Βερβιλιu γλυχερος πονος, ηθεσι μιγνυς
      <lb/>Οπλα τ, κι τερπνην ουασιν ιςτορίαν.</orig>
    <reg resp="#Alain Legros">eis to auto.
      <lb/>Dzésetai Berbiliu ulucheros ponos, éthesi miunus
      <lb/>Opla t, ki terpnen ouasin istorian.
    </reg>
  </choice>
</foreign>
```

1) @resp devrait pointer (je suppose) sur une personne (Alain Legros) et non pas sur deux.
Donc mieux convertir cela en @resp="#AL" ou @resp="#AlainLegros"

2) le <orig>, evidemment, est en Grec, comme l'indique le xml:lang. Mais je ne reconnais pas la langue du <reg> -- ca ne peut pas etre du grec puisque c'est écrit en lettres latines!

Il faut au moins preciser une autre @xml:lang pour le <reg> n'est-ce pas?

J'ai vu plusieurs exemplaires comme ceci:

```
<supplied resp="#MD #IN40413">les</supplied>
```

ce qui devrait indiquer que la lecture "les" soit fournie par 2 agences "#MD" et "#IN40413".

MD est bien défini comme Mathieu Duboc, mais ne trouvant aucune définition de "IN40413" je me permet de spéculer que ça serait l'identifiant non pas d'une personne mais d'une source. En quel cas, on pourrait changer ça en

```
<supplied resp="#MD" source="#IN40413">les</supplied>
```

(avec une définition qqpart du "IN40413" bien sûr)

La fonction de <surrogates> est de documenter des versions (par ex, photographique, imprime, numérisé) du manuscrit qu'on est en train de cataloguer.

Un usage comme ceci:

```
<surrogates>
  <bibl>
    <relatedItem type="original">
      <ref target="#THIA_63_1">Bibliothèques virtuelles humanistes</ref>
    </relatedItem>
  </bibl>
</surrogates>
```

n'a aucun sens. Le lien avec l'original (le ms THIA 63 1) est déjà indiqué par le msDesc/msIdentifier. Ou est-ce que j'ai mal compris?

Les codes suivants sont utilisees comme valeur d'un attribut xml:id plus qu'une fois, ce qui n'est pas permis. Dans les 4 cas suivants, il s'agit d'un <bibl> et d'un <msIdentifier> dans le meme fichier, decrivant le meme objet.

RES_Y2_2169

RESP_Y2_1349

FR_1513

Gordon_1563_R66

La valeur B343_1 est utilise 3 fois, une fois sur un <msIdentifier> dans un fichier, et les 2 autres fois sur un <bibl> dans 2 fichiers divers.

"Protestant icy devant mon <persName key="#locm_helicon">Helicon</persName> en"

Helicon n'est pas un <persName>. Pareil pour les 2 suivants:

- mon <persName key="#locm_parnasse">Parnase</persName> versé à l'escole
d'<persName key="#persm_apollon">Apollo</persName>, &

- <lb rend="hyphen"/>ste de <persName key="#loc_rouillac">Rouillac</persName>, &
ayans ja par dix ans pedé

Et stp, enleve le blanc exigue dans le suivant :

<persName key="#pers_ pedroalvarezcabral">Pietre Aliares</persName>, & ne scay
combien

L'element <castList> est concu pour encoder la liste des personnages telle qu'elle apparait au debut (ou a la fin) d'un texte. Vous en utilisez ainsi dans quelques textes.

Mais il y a aussi 2 autres textes (et), ou cet element est utilise aussi pour encoder la liste de personnages "sur scene" au debut d'une scene, par ex. A mon avis, il serait mieux de ne pas confondre ces 2 types d'objet. Le deuxieme cas, a mon avis, serait plutot un type de <stage>.

Je souhaiterais par contre ajouter un <castList type="synthetic"> ou pareil, pour regrouper tous les roles identifies dans un texte a travers les valeurs de @who.

Dans <respStmt> on peut mettre <name> ou <persName> avec <resp>.

Au corpus actuel, j'ai retrouvé 280 exemplaires de <name> vs 99 pour <persName>

Je propose donc de transformer tous les respStmt/persName en respStmt/name (ce qui simplifie un peu les choses...)

Partout dans B759999999_Y2_2162_tei

il y a des erreurs d'expansion comme par ex

```
<said who="#trouillogan">
  <name>
    <choice>
      <abbr>TR.</abbr>
      <expn>PANURGE</expn>
    </choice>
  </name> Il me
  <b/>l'est advis.</said>
```

Pour les textes manuscrits, on fournit un <msDesc>, OK. Mais je viens de m'apercevoir que dans ces <msDesc>, c'est le <msIdentifier> qui porte un attribut @xml:id.

Je propose de le déplacer sur le <msDesc>: la valeur xml:id (par ex "Gordon_1536_R656") sert à identifier le manuscrit, non pas son identifiant!